



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/816,471	04/01/2004	Michael A. Wolf	AZULP003	8395
21912	7590	03/29/2006	EXAMINER	
VAN PELT, YI & JAMES LLP 10050 N. FOOTHILL BLVD #200 CUPERTINO, CA 95014			IWASHKO, LEV	
			ART UNIT	PAPER NUMBER
			2186	

DATE MAILED: 03/29/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No.	Applicant(s)	
	10/816,471	WOLF ET AL.	
	Examiner	Art Unit	
	Lev I. Iwashko	2186	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 01 April 2004.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-38 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-38 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 01 April 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All   b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

**DETAILED ACTION**

***Claim Rejections - 35 USC § 102***

1. The following are quotations of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

2. Claims 1-38 are rejected under U.S.C. 102(e) as being anticipated by Hagggar et al. (US PGPub 2002/0091904 A1).

Claim 1. A method for managing memory, comprising: *(Section 0017, lines 1-3 – state the following: “The present invention is described herein in the context of providing improved management of a memory pool”)*

- maintaining a memory pool; *(Title – States the following: “Methods, systems, and computer program products for memory pool management”)*
- specifying a specified amount of memory in the memory pool for allocation; *(Section 0035, lines 1-8 - State the following: “If there is memory available in the heap 34 to grant the storage block request as determined by the heap memory allocation module 38 at block 64, then*

*the heap memory allocation module 38 grows the size of the handle sub-pool 44 at block 66. Operations continue at block 62 where the heap memory allocation module 38 allocates memory from the handle sub-pool 44 to accommodate the heap memory request as discussed hereinabove”)*

- and requesting a process to release a requested amount of memory in the memory pool. *(Section 0038, lines 5-9 – State the following: “By allocating handles and storage blocks in separate, variable sized sub-pools, each data type may be allowed to consume and release heap 34 memory in accordance with the requirements of a particular application”)*

Claim 2. A method for managing memory as recited in Claim 1, wherein the process operates in a garbage-collected environment. *(Section 0039, lines 11-13 – State the following: “At block 82, the heap memory allocation module 38 determines whether garbage collection has been run for the present heap memory request”)*

Claim 3. A method for managing memory as recited in Claim 1, wherein the process is a Java process. *(Section 0025, lines 1-15 – State the following: “The heap 34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the temporary storage of data and/or information whose existence or size may not be able to be determined until the program(s) execute. In accordance with embodiments of the present invention, the heap 34 is configured with two, variable size sub-pools: a handle sub-pool 44 and a storage block sub-pool 46. The handle sub-pool 44 contains one or more handles, which are respectively associated with storage blocks in the storage block sub-pool, and may be used to provide indirect access to the storage blocks as discussed hereinabove. The storage block sub-pool 46 contains one or more storage blocks that may be respectively allocated, for example, to object(s) and/or for the storage of data or other information”)*

- Claim 4. A method for managing memory as recited in Claim 1, wherein the process is a Java program. *(Section 0027, lines 1-6 – State the following: “Finally, the Java application program(s) 36 represent any program or executable code, including, for example, the operating system 28, driver programs, utility programs, and traditional application programs (i.e., programs designed to assist in the performance of a specific task), that may cooperate with the JVM 32 to use the heap 34”)*
- Claim 5. A method for managing memory as recited in Claim 1, wherein the process is a memory releasing process; *(Section 0007, lines 14-17 – State the following: “By allocating handles and storage blocks in separate, variable sized sub-pools, each data type may be allowed to consume and release memory in accordance with the requirements of a particular application”)*
- and further comprising allocating the specified amount of memory to a memory-requesting process. *(Section 0035, lines 5-8 – State the following: “Operations continue at block 62 where the heap memory allocation module 38 allocates memory from the handle sub-pool 44 to accommodate the heap memory request as discussed hereinabove”)*
- Claim 6. A method for managing memory as recited in Claim 1, wherein the memory pool includes reserved memory. *(Section 0025, lines 1-5 – State the following: “The heap 34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the temporary storage of data and/or information whose existence or size may not be able to be determined until the program(s) execute.”)*
- Claim 7. A method for managing memory as recited in Claim 1, wherein the memory pool includes memory owned by a plurality of processes. *(Section 0022, lines 1-13 – State the following: “As shown in FIG. 2, the memory 24 may hold at least four major categories of software and data: the operating system 28, the Java Virtual Machine (JVM) 32, the heap 34, and the Java application program(s) 36. The operating system 28 controls the*

*operation of the computer system. In particular, the operating system 28 may manage the computer system's resources and may coordinate execution of programs by the processor 22. The JVM 32, as described hereinabove, comprises software programs that emulate the processor 22 to allow the Java application programs 36 to execute on the processor 22 for a variety of different types of operating system types (e.g., UNIX, Windows NT, etc.)." The fact that there are multiple applications denotes that there are multiple processes. Section 0007, lines 14-17 – State the following: "By allocating handles and storage blocks in separate, variable sized sub-pools, each data type may be allowed to consume and release memory in accordance with the requirements of a particular application")*

- Claim 8. A method for managing memory as recited in Claim 1, wherein the memory pool includes a plurality of subpools. *(Section 0032, lines 4-7 – State the following: "Referring now to FIG. 4, operations begin at blocks 52 and 54 where a variable size storage block sub-pool 46 and a variable size handle sub-pool 44 are provided in the heap 34")*
- Claim 9. A method for managing memory as recited in Claim 1, further comprising determining that the specified amount of memory is required for allocation. *(Section 0034, lines 7-11 – State the following: "Specifically, the heap memory allocation module 38 determines whether the size of both the handle sub-pool 44 and the storage block sub-pool 46 may be increased to accommodate the storage block request without exceeding the size of the heap")*
- Claim 10. A method for managing memory as recited in Claim 1, further comprising receiving a memory request from a requesting process for the specified amount of memory and determining that the specified amount of memory is required for allocation. *(Section 0037, lines 5-10 – To grant the heap memory request, the heap memory allocation module 38 increases the size of the storage block sub-pool 46 based on the size of the requested storage*

- block and may also increase the size of the handle sub-pool 44 if the handle sub-pool 44 does not contain any unused memory)*
- Claim 11. A method for managing memory as recited in Claim 1, wherein the process is a releasing process; *(Section 0038, lines 5-9 – State the following: “By allocating handles and storage blocks in separate, variable sized sub-pools, each data type may be allowed to consume and release heap 34 memory in accordance with the requirements of a particular application”)*
- and further comprising monitoring a monitored process and determining that the monitored process requires additional memory. *(Section 0008, lines 1-5 – State the following: “In accordance with further embodiments of the present invention, if there is insufficient memory in the memory pool to grant a storage block request, the memory pool may be examined to determine if there are any storage blocks that are no longer in use”)*
- Claim 12. A method for managing memory as recited in Claim 1, wherein the process is a releasing process; *(Section 0038, lines 5-9 – State the following: “By allocating handles and storage blocks in separate, variable sized sub-pools, each data type may be allowed to consume and release heap 34 memory in accordance with the requirements of a particular application.” To “examine” is synonymous to “monitor”)*
- and further comprising monitoring a monitored process and detecting a rate of garbage collection for the monitored process. *(Section 0008, lines 5-12 – “The operation of examining a region in memory and reclaiming unused portions for reuse is often referred to as “garbage collection.” Note that garbage collection may also be implemented as a background process to be performed on a periodic basis. If one or more storage blocks are detected as being no longer in use, these storage block(s) and their associated handle(s) may be deallocated.” To “examine” is synonymous to “monitor”)*

- Claim 13. A method for managing memory as recited in Claim 1, wherein requesting a process to release a requested amount of memory in the memory pool includes selecting the process from a plurality of processes based on status information. *(Section 0025, lines 1-15 – State the following: “The heap 34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the temporary storage of data and/or information whose existence or size may not be able to be determined until the program(s) execute. In accordance with embodiments of the present invention, the heap 34 is configured with two, variable size sub-pools: a handle sub-pool 44 and a storage block sub-pool 46. The handle sub-pool 44 contains one or more handles, which are respectively associated with storage blocks in the storage block sub-pool, and may be used to provide indirect access to the storage blocks as discussed hereinabove. The storage block sub-pool 46 contains one or more storage blocks that may be respectively allocated, for example, to object(s) and/or for the storage of data or other information.” Section 0037, lines 12-13 – State the following: “Each handle contains a pointer and other fixed size information; therefore, when the handle sub-pool 44 is grown, the handle sub-pool's size increases by a fixed amount”)*
- Claim 14. A method for managing memory as recited in Claim 1, wherein requesting a process to release a requested amount of memory in the memory pool includes making a request via a system call. *(Section 0019, lines 10-14 – State the following: “In the context of this document, a computer-usable or computer-readable medium may be any medium that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device”)*
- Claim 15. A method for managing memory as recited in Claim 1, wherein requesting a process to release a requested amount of memory in the memory pool includes making a request via an inter-process communication protocol. *(Section 0018, lines 1-14 – State the following: “The present invention*



*may be embodied as methods, systems, and/or computer program products. Accordingly, the present invention may be embodied in software (including firmware, resident software, micro-code, etc.). Furthermore, the present invention may take the form of a computer program product on a computer-usable or computer-readable storage medium having computer-usable or computer readable program code embodied in the medium for use by or in connection with an instruction execution system. In the context of this document, a computer-usable or computer-readable medium may be any medium that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device")*

Claim 16. A method for managing memory as recited in Claim 1, wherein the specified amount of memory is approximately equal to the requested amount of memory. (Section 0034, lines 1-11 – State the following: “If, however, the heap memory allocation module 38 determines at block 58 that there is not sufficient memory available in the handle sub-pool 44 to provide a handle for the storage block request, then the heap memory allocation module makes a determination at block 64 whether there is sufficient memory in the heap 34 to grant the storage block request. Specifically, the heap memory allocation module 38 determines whether the size of both the handle sub-pool 44 and the storage block sub-pool 46 may be increased to accommodate the storage block request without exceeding the size of the heap”)

Claim 17. A method for managing memory ms recited in Claim 1, wherein the requested amount of memory in the memory pool is freeable memory. (Section 0024, lines 4-6 – State the following: “In particular, storage blocks and handles that are no longer in use are identified and are deallocated to free up memory in the heap 34.” If the memory is freed, then it must have been freeable)

- Claim 18. A method for managing memory as recited in Claim 1, further comprising refilling a subpool of the memory pool with the requested amount of memory released by the process. *(Section 0038, lines 9-15 – State the following: “New handles and storage blocks may be allocated until the heap 34 is filled. Inasmuch as the heap is shared between handles and storage blocks, the problem of having a fill heap even though space is still available in one of the sub-pools may be avoided. In addition, the improved utilization of the heap 34 may allow for a smaller heap size, which conserves memory”)*
- Claim 19. A computer program product for managing memory, the computer program product being embodied in a computer readable medium and comprising computer instructions for: *(Section 0019, lines 1-19 – State the following: “The computer-usable or computer-readable medium may be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples (a nonexhaustive list) of the computer usable or computer-readable medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, and a portable compact disc read-only memory (CD-ROM). Note that the computer-usable or computer-readable medium could even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via, for instance, optical scanning of the paper or other medium, then compiled, interpreted or otherwise processed in a suitable manner if necessary, and then stored in a computer memory.” Title – States the following: “Methods, systems, and computer program products for memory pool management”)*

- maintaining a memory pool; (*Title – States the following: “Methods, systems, and computer program products for memory pool management”*)
- specifying a specified amount of memory in the memory pool for allocation; (*Section 0035, lines 1-8 - State the following: “If there is memory available in the heap 34 to grant the storage block request as determined by the heap memory allocation module 38 at block 64, then the heap memory allocation module 38 grows the size of the handle sub-pool 44 at block 66. Operations continue at block 62 where the heap memory allocation module 38 allocates memory from the handle sub-pool 44 to accommodate the heap memory request as discussed hereinabove”*)
- and requesting a process to release a requested amount of memory in the memory pool. (*Section 0038, lines 5-9 – State the following: “By allocating handles and storage blocks in separate, variable sized sub-pools, each data type may be allowed to consume and release heap 34 memory in accordance with the requirements of a particular application”*)

- Claim 20. A memory management system, comprising: (*Section 0017, lines 1-3 – state the following: “The present invention is described herein in the context of providing improved management of a memory pool”*)
- a memory pool; (*Figure 3 – Shows a diagram of a memory pool*)
  - a processor coupled to the memory pool, configured to: (*Figure 2, components 22, 24 and 24 – Shows how the processor is coupled to the memory pool*)
  - maintain the memory pool; (*Title – States the following: “Methods, systems, and computer program products for memory pool management”*)
  - specifying a specified amount of memory in the memory pool for allocation; (*Section 0035, lines 1-8 - State the following: “If there is*

*memory available in the heap 34 to grant the storage block request as determined by the heap memory allocation module 38 at block 64, then the heap memory allocation module 38 grows the size of the handle sub-pool 44 at block 66. Operations continue at block 62 where the heap memory allocation module 38 allocates memory from the handle sub-pool 44 to accommodate the heap memory request as discussed hereinabove”)*

- *and request a process to release a requested amount of memory in the memory pool. (Section 0038, lines 5-9 – State the following: “By allocating handles and storage blocks in separate, variable sized sub-pools, each data type may be allowed to consume and release heap 34 memory in accordance with the requirements of a particular application”)*

Claim 21. A method for managing memory, comprising: (Section 0017, lines 1-3 – state the following: “The present invention is described herein in the context of providing improved management of a memory pool”)

- *maintaining a memory pool; (Title – States the following: “Methods, systems, and computer program products for memory pool management”)*
- *receiving status information from a plurality of processes; and managing memory among the plurality of processes using the status information. (Section 0025, lines 1-15 – State the following: “The heap 34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the temporary storage of data and/or information whose existence or size may not be able to be determined until the program(s) execute. In accordance with embodiments of the present invention, the heap 34 is configured with two, variable size sub-pools: a handle sub-pool 44 and a storage block sub-pool 46. The handle sub-pool 44 contains one or more handles, which are respectively associated with storage*

*blocks in the storage block sub-pool, and may be used to provide indirect access to the storage blocks as discussed hereinabove. The storage block sub-pool 46 contains one or more storage blocks that may be respectively allocated, for example, to object(s) and/or for the storage of data or other information.” Section 0037, lines 12-13 – State the following: “Each handle contains a pointer and other fixed size information; therefore, when the handle sub-pool 44 is grown, the handle sub-pool’s size increases by a fixed amount”)*

Claim 22. A method for managing memory as recited in Claim 21, wherein managing memory among the plurality of processes includes allocating memory to one of the plurality of processes. (Section 0022, lines 1-13 – State the following: “As shown in FIG. 2, the memory 24 may hold at least four major categories of software and data: the operating system 28, the Java Virtual Machine (JVM) 32, the heap 34, and the Java application program(s) 36. The operating system 28 controls the operation of the computer system. In particular, the operating system 28 may manage the computer system’s resources and may coordinate execution of programs by the processor 22. The JVM 32, as described hereinabove, comprises software programs that emulate the processor 22 to allow the Java application programs 36 to execute on the processor 22 for a variety of different types of operating system types (e.g., UNIX, Windows NT, etc.).” The fact that there are multiple applications denotes that there are multiple processes. Section 0007, lines 14-17 – State the following: “By allocating handles and storage blocks in separate, variable sized sub-pools, each data type may be allowed to consume and release memory in accordance with the requirements of a particular application”)

Claim 23. A method for managing memory as recited in Claim 21, wherein managing memory among the plurality of processes includes requesting one of the plurality of processes to release memory. (Section 0007, lines 14-17 – State the following: “By allocating handles and storage blocks in

*separate, variable sized sub-pools, each data type may be allowed to consume and release memory in accordance with the requirements of a particular application”)*

Claim 24. A method for managing memory as recited in Claim 21, wherein the status information includes status of freeable memory. (Section 0024, lines 1-13 and Section 0025, lines 1-15 – State the following: “The garbage collection module 42 may be configured to automatically recover the heap 34 when the heap 34 becomes full preventing new storage blocks and handles from being allocated therefrom. In particular, storage blocks and handles that are no longer in use are identified and are deallocated to free up memory in the heap 34. Storage blocks that remain in use may be moved to consolidate the free memory into larger blocks. In accordance with various alternative embodiments of the invention, the garbage collection module 42 may un periodically to free up memory in the heap and/or the garbage collection module 42 may run upon failure of the heap memory allocation module 38 to fulfill a heap memory request from an application.” “The heap 34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the temporary storage of data and/or information whose existence or size may not be able to be determined until the program(s) execute. In accordance with embodiments of the present invention, the heap 34 is configured with two, variable size sub-pools: a handle sub-pool 44 and a storage block sub-pool 46. The handle sub-pool 44 contains one or more handles, which are respectively associated with storage blocks in the storage block sub-pool, and may be used to provide indirect access to the storage blocks as discussed hereinabove. The storage block sub-pool 46 contains one or more storage blocks that may be respectively the storage of data or other information”)

Claim 25. A method for managing memory as recited in Claim 21, wherein the status information includes efficiency of the process's garbage collector. (Section

*0010, lines 4-7 – State the following: “Because of the improved memory pool utilization provided by the present invention, garbage collection may be run less frequently to free up memory in the memory pool”)*

Claim 26. A method for managing memory as recited in Claim 21, further comprising requesting status information. *(Section 0037, lines 1-26 – State the following: “In view of the foregoing, if a heap memory request is based on the creation of a new object, for example, then the requested storage block is allocated to the new object and the new object may be indirectly accessed through the handle associated with the requested storage block. To grant the heap memory request, the heap memory allocation module 38 increases the size of the storage block sub-pool 46 based on the size of the requested storage block and may also increase the size of the handle sub-pool 44 if the handle sub-pool 44 does not contain any unused memory (i.e., deallocated handle(s)), which may be used to provide the handle for the instant storage block request. Each handle contains a pointer and other fixed size information; therefore, when the handle sub-pool 44 is grown, the handle sub-pool's size increases by a fixed amount. By contrast, the storage blocks may vary in size; therefore, the size of the storage block sub-pool increases according to the size of the requested storage block. In particular embodiments of the present invention in which the handle sub-pool 44 is disposed at one end of the heap 34 and the storage block sub-pool 46 is disposed at the other end of the heap 34 as shown in FIG. 3, the end of the handle sub-pool 44 is extended towards the end address of the heap 34 based on the size of the handle and the end of the storage block sub-pool 46 is extended towards the beginning address of the heap 34 based on the size of the requested storage block”)*

Claim 27. A method for managing memory as recited in Claim 21, wherein the status information is sent along with a memory allocation request. *(Section 0025, lines 1-15 and Section 0026, lines 1-14 – State the following: “The heap*

*34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the temporary storage of data and/or information whose existence or size may not be able to be determined until the program(s) execute. In accordance with embodiments of the present invention, the heap 34 is configured with two, variable size sub-pools: a handle sub-pool 44 and a storage block sub-pool 46. The handle sub-pool 44 contains one or more handles, which are respectively associated with storage blocks in the storage block sub-pool, and may be used to provide indirect access to the storage blocks as discussed hereinabove. The storage block sub-pool 46 contains one or more storage blocks that may be respectively allocated, for example, to object(s) and/or for the storage of data or other information.” “FIG. 3 illustrates an exemplary configuration of the heap 34, in accordance with embodiments of the present invention. As shown in FIG. 3, the variable size handle sub-pool 44 begins at a first end (i.e., the beginning address) of the heap 34 and extends towards a second end (i.e., the end address) of the heap 34 as the heap memory allocation module 38 allocates new handles. Conversely, the variable size storage block sub-pool 46 begins at the second end of the heap 34 and extends towards the first end of the heap 34 as the heap memory allocation module 38 allocates new storage blocks. Note that the individual storage blocks may vary in size, but the individual handles are all a fixed size because they contain fixed size pointers (i.e., addresses) and fixed size information about the respective storage blocks”)*

- Claim 28. A method for managing memory as recited in Claim 21, wherein the status information is received periodically. (Section 0024, lines 8-13 – State the following: “In accordance with various alternative embodiments of the invention, the garbage collection module 42 may un periodically to free up memory in the heap and/or the garbage collection module 42 may run



*upon failure of the heap memory allocation module 38 to fulfill a heap memory request from an application.”)*

- Claim 29. A computer program product for managing memory, the computer program product being embodied in a computer readable medium and comprising computer instructions for: *(Section 0019, lines 1-19 – State the following: “The computer-usable or computer-readable medium may be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples (a nonexhaustive list) of the computer usable or computer-readable medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, and a portable compact disc read-only memory (CD-ROM). Note that the computer-usable or computer-readable medium could even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via, for instance, optical scanning of the paper or other medium, then compiled, interpreted or otherwise processed in a suitable manner if necessary, and then stored in a computer memory.” Title – States the following: “Methods, systems, and computer program products for memory pool management”)*
- maintaining a memory pool; *(Title – States the following: “Methods, systems, and computer program products for memory pool management”)*
  - receiving status information from a plurality of processes; and managing memory among the plurality of processes using the status information. *(Section 0025, lines 1-15 – State the following: “The heap 34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the*

*temporary storage of data and/or information whose existence or size may not be able to be determined until the program(s) execute. In accordance with embodiments of the present invention, the heap 34 is configured with two, variable size sub-pools: a handle sub-pool 44 and a storage block sub-pool 46. The handle sub-pool 44 contains one or more handles, which are respectively associated with storage blocks in the storage block sub-pool, and may be used to provide indirect access to the storage blocks as discussed hereinabove. The storage block sub-pool 46 contains one or more storage blocks that may be respectively allocated, for example, to object(s) and/or for the storage of data or other information.” Section 0037, lines 12-13 – State the following: “Each handle contains a pointer and other fixed size information; therefore, when the handle sub-pool 44 is grown, the handle sub-pool's size increases by a fixed amount”)*

- Claim 30. A memory management system, comprising: (Section 0017, lines 1-3 – state the following: “The present invention is described herein in the context of providing improved management of a memory pool”)
- a memory pool; (Figure 3 – Shows a diagram of a memory pool)
  - a processor coupled to the memory pool, configured to: (Figure 2, components 22, 24 and 24 – Shows how the processor is coupled to the memory pool)
  - maintain a memory pool; (Title – States the following: “Methods, systems, and computer program products for memory pool management”)
  - receive status information from a plurality of processes; and manage memory among the plurality of processes using the status information. (Section 0025, lines 1-15 – State the following: “The heap 34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the temporary storage of data and/or information whose existence or size may not be able to be

*determined until the program(s) execute. In accordance with embodiments of the present invention, the heap 34 is configured with two, variable size sub-pools: a handle sub-pool 44 and a storage block sub-pool 46. The handle sub-pool 44 contains one or more handles, which are respectively associated with storage blocks in the storage block sub-pool, and may be used to provide indirect access to the storage blocks as discussed hereinabove. The storage block sub-pool 46 contains one or more storage blocks that may be respectively allocated, for example, to object(s) and/or for the storage of data or other information.” Section 0037, lines 12-13 – State the following: “Each handle contains a pointer and other fixed size information; therefore, when the handle sub-pool 44 is grown, the handle sub-pool's size increases by a fixed amount”)*

- Claim 31. A method for managing memory, comprising: (Section 0017, lines 1-3 – state the following: “The present invention is described herein in the context of providing improved management of a memory pool”)
- maintaining a memory pool; (Title – States the following: “Methods, systems, and computer program products for memory pool management”)
  - determining that an amount of memory in the memory pool is required for allocation; (Section 0035, lines 1-8 - State the following: “If there is memory available in the heap 34 to grant the storage block request as determined by the heap memory allocation module 38 at block 64, then the heap memory allocation module 38 grows the size of the handle sub-pool 44 at block 66. Operations continue at block 62 where the heap memory allocation module 38 allocates memory from the handle sub-pool 44 to accommodate the heap memory request as discussed hereinabove”)
  - and allocating the amount of memory to a process. (Section 0033, lines 1-13 – State the following: “Referring now to FIGS. 5A and 5B,

*exemplary operations for processing a storage block request from a program will be described hereafter. At block 56, the heap memory allocation module 38 receives a heap memory request from a Java application program 36. The heap memory allocation module 38 makes a determination at block 58 whether there is unused memory (i.e., deallocated handle(s)) available in the handle sub-pool 44 to accommodate the heap memory request. If there is memory available in the handle sub-pool 44 to provide a handle to accommodate the storage block request, then the heap memory allocation module 38 allocates this memory for the handle at block 62")*

- Claim 32. A method for managing memory as recited in Claim 31, wherein determining that an amount of memory in the memory pool is required for allocation includes determining a request priority. *(Section 0043, lines 7-13 – State the following: “It should also be noted that in some alternative implementations, the functions noted in the blocks may occur out of the order noted in FIGS. 4-6. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may be executed in the reverse order, depending on the functionality involved”)*
- Claim 33. A method for managing memory as recited in Claim 31, wherein determining that an amount of memory in the memory pool is required for allocation includes receiving a request having an urgency level and determining a request priority based on the urgency level. *(Section 0043, lines 7-13 – State the following: “It should also be noted that in some alternative implementations, the functions noted in the blocks may occur out of the order noted in FIGS. 4-6. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may be executed in the reverse order, depending on the functionality involved.” The functionality involved is like urgency)*
- Claim 34. A method for managing memory as recited in Claim 31, wherein determining that an amount of memory in the memory pool is required for

allocation includes determining whether the process includes memory collateral. (Section 0033, lines 5-7 – State the following: “The heap memory allocation module 38 makes a determination at block 58 whether there is unused memory.” Memory collateral refers to unused memory.)

Claim 35. A method for managing memory as recited in Claim 31, wherein allocating the amount of memory to a process includes selecting the amount of memory from a subpool of the memory pool. (Section 0025, lines 1-15 – State the following: “The heap 34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the temporary storage of data and/or information whose existence or size may not be able to be determined until the program(s) execute. In accordance with embodiments of the present invention, the heap 34 is configured with two, variable size sub-pools: a handle sub-pool 44 and a storage block sub-pool 46. The handle sub-pool 44 contains one or more handles, which are respectively associated with storage blocks in the storage block sub-pool, and may be used to provide indirect access to the storage blocks as discussed hereinabove. The storage block sub-pool 46 contains one or more storage blocks that may be respectively allocated, for example, to object(s) and/or for the storage of data or other information.” Section 0037, lines 12-13 – State the following: “Each handle contains a pointer and other fixed size information; therefore, when the handle sub-pool 44 is grown, the handle sub-pool's size increases by a fixed amount”)

Claim 36. A method for managing memory as recited in Claim 31, wherein allocating the amount of memory to a process includes selecting the amount of memory from a subpool of the memory pool; (Section 0025, lines 1-15 – State the following: “The heap 34 may be configured as a portion of the memory 24 that is reserved for use by the Java application program(s) 36 for the temporary storage of data and/or information whose existence or size may not be able to be determined until the

*program(s) execute. In accordance with embodiments of the present invention, the heap 34 is configured with two, variable size sub-pools: a handle sub-pool 44 and a storage block sub-pool 46. The handle sub-pool 44 contains one or more handles, which are respectively associated with storage blocks in the storage block sub-pool, and may be used to provide indirect access to the storage blocks as discussed hereinabove. The storage block sub-pool 46 contains one or more storage blocks that may be respectively allocated, for example, to object(s) and/or for the storage of data or other information.” Section 0037, lines 12-13 – State the following: “Each handle contains a pointer and other fixed size information; therefore, when the handle sub-pool 44 is grown, the handle sub-pool's size increases by a fixed amount”)*

- *further comprising refilling the subpool with released memory by the process. (Section 0038, lines 9-15 – State the following: “New handles and storage blocks may be allocated until the heap 34 is filled. Inasmuch as the heap is shared between handles and storage blocks, the problem of having a fill heap even though space is still available in one of the sub-pools may be avoided. In addition, the improved utilization of the heap 34 may allow for a smaller heap size, which conserves memory”)*

Claim 37. A computer program product for managing memory, the computer program product being embodied in a computer readable medium and comprising computer instructions for: (Section 0019, lines 1-19 – State the following: “The computer-usable or computer-readable medium may be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples (a nonexhaustive list) of the computer usable or computer-readable medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a random access memory (RAM), a read-only memory

*(ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, and a portable compact disc read-only memory (CD-ROM). Note that the computer-usable or computer-readable medium could even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via, for instance, optical scanning of the paper or other medium, then compiled, interpreted or otherwise processed in a suitable manner if necessary, and then stored in a computer memory.” Title – States the following:*

*“Methods, systems, and computer program products for memory pool management”)*

- *maintaining a memory pool; (Title – States the following: “Methods, systems, and computer program products for memory pool management”)*
- *determining that an amount of memory in the memory pool is required for allocation; (Section 0035, lines 1-8 - State the following: “If there is memory available in the heap 34 to grant the storage block request as determined by the heap memory allocation module 38 at block 64, then the heap memory allocation module 38 grows the size of the handle sub-pool 44 at block 66. Operations continue at block 62 where the heap memory allocation module 38 allocates memory from the handle sub-pool 44 to accommodate the heap memory request as discussed hereinabove”)*
- *and allocating the amount of memory to a process. (Section 0033, lines 1-13 – State the following: “Referring now to FIGS. 5A and 5B, exemplary operations for processing a storage block request from a program will be described hereafter. At block 56, the heap memory allocation module 38 receives a heap memory request from a Java application program 36. The heap memory allocation module 38 makes a determination at block 58 whether there is unused memory (i.e., deallocated handle(s)) available in the handle sub-pool 44 to*

*accommodate the heap memory request. If there is memory available in the handle sub-pool 44 to provide a handle to accommodate the storage block request, then the heap memory allocation module 38 allocates this memory for the handle at block 62")*

- Claim 38. A memory management system, comprising: (Section 0017, lines 1-3 – state the following: “The present invention is described herein in the context of providing improved management of a memory pool”)
- a memory pool; (Figure 3 – Shows a diagram of a memory pool)
  - a processor coupled to the memory pool, configured to: (Figure 2, components 22, 24 and 24 – Shows how the processor is coupled to the memory pool)
  - maintain a memory pool; (Title – States the following: “Methods, systems, and computer program products for memory pool management”)
  - determine that an amount of memory in the memory pool is, required for allocation; (Section 0035, lines 1-8 - State the following: “If there is memory available in the heap 34 to grant the storage block request as determined by the heap memory allocation module 38 at block 64, then the heap memory allocation module 38 grows the size of the handle sub-pool 44 at block 66. Operations continue at block 62 where the heap memory allocation module 38 allocates memory from the handle sub-pool 44 to accommodate the heap memory request as discussed hereinabove”)
  - and allocate the amount of memory to a process. (Section 0033, lines 1-13 – State the following: “Referring now to FIGS. 5A and 5B, exemplary operations for processing a storage block request from a program will be described hereafter. At block 56, the heap memory allocation module 38 receives a heap memory request from a Java application program 36. The heap memory allocation module 38 makes a determination at block 58 whether there is unused memory



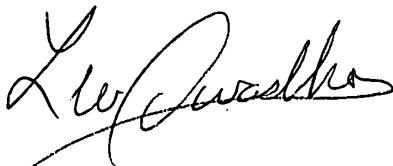
*(i.e., deallocated handle(s)) available in the handle sub-pool 44 to accommodate the heap memory request. If there is memory available in the handle sub-pool 44 to provide a handle to accommodate the storage block request, then the heap memory allocation module 38 allocates this memory for the handle at block 62")*

**Conclusion**

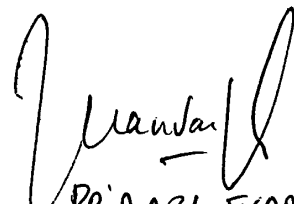
3. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lev I. Iwashko whose telephone number is (571)272-1658. The examiner can normally be reached on 9 Hours Schedule), from 8-4PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt Kim can be reached on (571)272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Lev Iwashko



PRIMARY EXAMINER  
GROUP 2100